

VIDEO STREAMING OVER THE PUBLIC INTERNET: MULTIPLE DESCRIPTION CODES AND ADAPTIVE TRANSPORT PROTOCOLS

Sergio D. Servetto*[†]

Klara Nahrstedt*

<http://www.ifp.uiuc.edu/~servetto/> <http://cairo.cs.uiuc.edu/>

*Dept. of Computer Science, [†]Beckman Institute,
Univ. of Illinois at Urbana-Champaign.
1304 W. Springfield Ave., Urbana, IL, 61801.

ABSTRACT

The problem of video transmission over IP networks is considered in this paper: we present the design of a complete system for the transmission of high-quality video signals over the public Internet. Our system involves (1) modifications to the standard Internet transport protocol (TCP), to make it suitable for the transport of delay-constrained traffic, and to gather channel state information; (2) the design of an error-resilient video coder; and (3) the design of a controller which uses channel state information provided by the modified protocol, to configure the error-resilient coder accordingly. Experimental studies performed on certain high-speed segments of the public Internet reveal that the proposed system is able to deliver video signals at bit rates in the range of 4-9 Mbits/sec, not only within fast LANs but also, and more surprisingly, even between sites in the US and Europe. That is, over existing public IP channels, our proposed system can deliver video at TV-broadcast quality.

1. INTRODUCTION

1.1. Problem Statement and Challenges

The main goal pursued in this paper is the design and implementation of a system to transmit in real time high-quality video signals over the public Internet. Our interest is in the development of techniques suitable for Video-On-Demand applications, such as WWW servers. Most video streaming systems that we are aware of typically operate at bit rates in the range of 40-50 Kbits/sec, mainly because a significant portion of Internet users are connected via dialup lines whose capacity is limited to at most 56 Kbits/sec (under the most optimistic assumptions). In this paper however, we consider the problem of *broadcast-quality* video transmission over IP channels, i.e., transmission of CCIR601 color signals at bit rates in the range 4-9 Mbits/sec. We are motivated to consider this problem for a number of reasons:

- We believe that the limitations on channel capacity imposed by “the last mile” are only temporary. New technologies like cable modems, digital subscriber loops, etc., promise the delivery of substantially higher data rates. Thus, from a research perspective, it makes sense to consider design issues for future systems free

from the very low bit rate constraint. Furthermore, video coding at very low bit rates is a problem that has received significant attention, for which a number of good solutions already exist (e.g., H.263).

- Free from the low bit rate constraint, new interesting technical issues arise. Specifically, if the bottleneck is not at the dialup line, where is it? Regarding the public Internet as a communication channel in the classical sense, what properties does it have? What is its capacity? How does capacity change over time, and what causes these changes? How can channel states be estimated? What are good coding schemes to transmit data over this channel?

It has been argued recently that “fifty years from now ... telecommunications will likely be dominated by packet data/voice transmitted over wide-area networks like the Internet ... the challenge will be to understand the network as a whole and to guarantee end-to-end quality of service” [2]. We believe the questions we pose are very meaningful, well aligned with this vision. Our main goal in this paper is to start to provide answers for them.

1.2. State-of-the-art Streaming Systems

The problem of sending video over IP has essentially two main components: video data compression, and design of communication protocols, as shown in Fig. 1.



Figure 1: Structure of a video streaming system.

1.2.1. A Coding-Only View of the Problem

One approach to this problem consists of designing a low bit rate coder, possibly protecting the resulting bit stream with channel codes [1, 5, 12], and using one of the standard Internet transport protocols to transmit the resulting data. If the source bit rate is low enough, then it is possible to use TCP, in which case no errors occur and therefore there is no need for channel codes. Otherwise, UDP is used with a constant packet injection rate, and low-redundancy channel codes are used to protect against infrequent lost packets.

The main drawback of this approach is that, essentially, it does not deal well with the time-varying nature of the channel. Besides, when using UDP, applications are responsible

This work was supported in part by the Ray Ozzie Fellowship of the Dept. of Computer Science at UIUC, and by the NSF under award CCR-96-23867 (CAREER) and grant #9624396.

for performing their own congestion avoidance and control: this may cause severe degradation of the overall network performance, due to potentially misbehaving sources.

1.2.2. A Networking-Only View of the Problem

An alternative approach to this problem consists of designing new transport protocols [3, 6], and using standard video coding algorithms. In this case, smart packetization schemes are used in order to minimize error propagation in the event of a packet loss (e.g., by gathering complete motion paths within a packet). Also, it is possible to retransmit packets selectively depending on, for example, whether a lost packet contains I-blocks or not. In this case, all flow control tasks are performed by the modified protocol.

The main drawback of this approach is that, since protocols cannot ensure error-free transmission, when errors do indeed occur, the quality of the decoded signals suffers severely because of lack of robustness in the coders used.

1.3. A Joint Source/Channel Coding Approach

Consider Fig. 1 again. One can easily recognize in that picture the structure of a communications system proposed by Shannon: there is a module in charge of removing redundancy in the source (the video coder), there is a module in charge of protecting the data against possible channel impairments (the transmission protocol), there is a channel (the network), and there are modules that invert the previous steps. Under this light, it is clear that the approaches previously described are essentially based on the Source/Channel Separation Theorem [4, Ch. 8]. And this is the first indicator of trouble: it is well known that the separation principle does *not* hold under delay constraints, under channel uncertainty, and for multiple access channels. In our approach instead, we propose to design a data compression algorithm *jointly* with the transmission protocol that will be used to move the resulting compressed stream, as illustrated in Fig. 2.

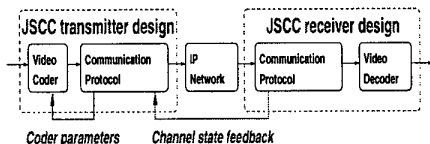


Figure 2: A Joint Source/Channel Coding (JSCC) view of the problem. In this approach, the interactions between the coder and the protocol are explicitly taken into account.

To accomplish our goal, there are essentially two issues that we need to consider: (a) we need to understand properties of the channel, to be able to design efficient methods for encoding and transmitting information across it; and (b) we need to design devices which can estimate and track channel state fluctuations, as well as convey this information in a useful manner to the encoder.

1.4. Main Contributions and Paper Organization

The main contributions presented in this work are:

- (a) A simple modification of TCP/IP (dubbed R-TCP), to make it suitable for the transport of delay-constrained data, and to gather channel state information.
- (b) The design of an algorithm for compressing video data, with the salient property that its performance degrades gracefully with the number of lost packets, but is independent of exactly *which* packets are lost.
- (c) The design of a rate control algorithm that “binds” together the protocol and the coder: parameters of the error-resilient coder are set as a function of channel state information gathered by the protocol.

A fundamental point we try to make in this paper is that there is an inherent mismatch between properties of state-of-the-art video coders and properties of existing transport protocols, and that these design tasks need to be addressed *jointly*. To the best of our knowledge, the interaction between coders and transport protocols had been considered before only by Garrett [7] and McCanne [11]. In both cases however, the solutions obtained are based on *layered* coders: as a result, communications systems designed using these coders are completely different from systems designed using coders like ours, in which all data packets are equally important in terms of decoded signal quality.

The rest of this paper is organized as follows. In Section 2 we present our proposed modifications to TCP, and in Section 3 we present an error-resilient video coder. In Section 4 we present the design of a controller which sets parameters of the coder as a function of the channel state information provided by the protocol. In Section 5 we present performance results. The paper concludes with Section 6.

2. R-TCP: REAL-TIME TCP

2.1. Mechanics of R-TCP

The modifications we introduce to standard TCP are:¹

Elimination of retransmissions. When the timer for a transmitted packet expires, instead of retransmitting this assumed lost packet (as specified by standard TCP), a fresh new packet is transmitted.

Partial acknowledgements. In TCP, an ack’ed data segment implies all previous segments are available at the receiver [17]. Since without retransmissions some packets will not be available, the meaning of ack messages is redefined to accommodate these error conditions.

RTP Headers. Timestamps, sequence numbers, etc., are added using the standard RTP protocol [13].

2.2. Hidden Markov Modeling of R-TCP Channels

In order for upper layers in the communication architecture to use information about channel states, the first requirement is that of a model that can be used to “describe” the channel to these higher layers. The development of such models for R-TCP connections is not a simple task: feedback in the flow control algorithm introduces memory and

¹For details on the standard TCP the reader is referred to [17], and specifically for details on the flow control technique to a classical paper of V. Jacobson [9].

time-dependencies in the service rate of the channel, which renders inappropriate the use of simple parameterizations based on classical models (e.g., exponential service rate).

Inspired by work on models for flat fading channels [18], a Hidden Markov Model (HMM) has been proposed to capture the statistical properties of TCP channels [14]:

- An R-TCP connection is regarded as a source that generates 1's and -1's at random times: 1's represent acknowledged and -1's unacknowledged packets.
- Time is quantized into intervals of length T such that, over any one interval $[t, t+T)$, at most one symbol can be generated.²
- A discrete-time process $(X_n)_{n \geq 0}$ is defined by $X_n = 0$, if no symbols were generated during $[nT, (n+1)T)$, else $X_n = 1, -1$; this process is modeled as an HMM.

For details on the construction of the model, parameter estimation techniques, motivation for the use of HMMs, etc., the reader is referred to [14]. What is of interest to us here is the fact that HMMs provide a compact and realistic description of the channel, one that we can use in the design of our JSCC algorithms.

3. MULTIPLE DESCRIPTION VIDEO CODING

In this section, we consider the problem of compressing a video signal into any number of mutually refinable packets. This section builds heavily on previous work on multiple description *image* coding, and hence due to space constraints is kept short: the reader is referred to [16] for details.

3.1. Temporal Dependencies in the Video Signal

The key feature that distinguishes video coding from still image coding is the presence of temporal dependencies among video frames which, if exploited correctly, lead to higher compression efficiency than that attainable by an efficient still image coder applied on each video frame separately. In order to take advantage of these temporal dependencies, the most popular techniques are (a) motion compensation, and (b) subband filtering along the time direction (known as 3D subbands). From a *pure* compression efficiency viewpoint, there is a general consensus in the research community that 3D subband coders do not perform as well as coders based on motion compensation.³

We claim that in the presence of a dirty channel, that consensus on the superiority of motion compensation techniques over 3D subband techniques ought to be revised. Motion compensation creates long-range dependencies among portions of the encoded bit streams: as a result, even a small amount of noise in the channel may lead to a total decoding failure, by means of error propagation. On the other hand, no such long-range dependencies exist

²For example, T could be taken as the amount of time it takes to execute a write operation on a socket at the transmitter.

³We want to emphasize that this observation relates *only* to pure compression efficiency. 3D subband coders are able to provide other functionalities that are much more difficult to provide using a coder based on motion compensation (e.g., scalability in bit rate, image size, and frame rate). In the design of a *complete system*, these are most important issues to take into account.

among 3D subband coefficients, and hence this representation *might* potentially perform better in noisy environments. Therefore, we do not feel we are incurring in any obvious suboptimality when we choose to design a robust coder based on 3D subbands. Furthermore, 3D subbands provide a natural framework in which to perform Unequal Error Protection (UEP): different subbands can be protected with different amounts of redundancy. Such UEP schemes become cumbersome when using motion compensation in the video coder.

3.2. Algorithms

Our error resilient video coder works as follows. It takes as input a color sequence in YUV format, of size 480 rows \times 704 columns \times 30 frames/sec, and blocks it into groups of pictures (GOPs) of 32 frames. Each color band is dealt with separately. A 5-level Haar decomposition is applied along the temporal direction, and then 5 levels of a 10-18 Daubechies wavelet decomposition are applied on each of the resulting temporal subbands. Then, each resulting spatio-temporal subband is compressed into a desired number of packets and with a desired amount of redundancy, using robust compression techniques from previous work [16].

4. RATE AND REDUNDANCY CONTROL

Whereas in classical –non error-resilient– source coders it is of interest to control over time the amount of bits at the output of the encoder (rate control), in JSCC systems like ours it is also necessary to consider the issue of how much robustness to errors is to be embedded over time (redundancy allocation). In this section we present the design of a controller which, as a function of the HMM describing the channel and the available observations, appropriately adjusts parameters of the encoder.

4.1. Rate Control

Given a sequence of N GOPs to transmit, our rate control problem consists of deciding how many packets to allocate to each GOP, in a way such that we transmit as many as possible for each GOP (to obtain better quality for the delivered signal), but without sending so many packets the transmission of any one GOP takes too long and prevents other GOPs from being delivered in time.

4.1.1. Problem Formulation

Let \mathcal{H} denote the HMM for the channel (with typical state and observation sequences $(s_k)_{k \geq 0}$ and $(b_k)_{k \geq 0}$), and let T_d denote the number of discrete-time units required to display a GOP.⁴ Consider also a function f defined by

$$f(u, \mathcal{H}) \triangleq \min \left\{ m : \sum_{k=0}^m |b_k| = u \right\},$$

which counts the number of clock ticks it takes to transmit u packets over the channel \mathcal{H} .

⁴ $T_d = \frac{1}{30} G_s / T$, where G_s is the number of frames in a GOP, and T is the time-quantization step.

As a function of this HMM, we construct a fully-observed dynamical system described by the following equations:

$$t_{n+1} = t_n + f(u_n, \mathcal{H}); \quad (1)$$

$$r_{n+1} = r_n + T_d; \quad (2)$$

$$\pi_{n+1} = \Pr(s_{t_n+f(u_n, \mathcal{H})} | \pi_n, b_{t_n}, \dots, b_{t_n+f(u_n, \mathcal{H})}) \quad (3)$$

with initial conditions $r_0 = t_0 = 0$, and $\pi_0 \sim$ the initial distribution of the HMM. These equations should be interpreted as follows. $(u_n)_{n \geq 0}$ is a sequence of control actions, which represent the number of packets allocated to the n -th GOP. t_n is the total number of clock ticks from the beginning of transmission until the transmission of the n -th GOP starts, and r_n is the number of clock ticks after which the display of the $n-1$ -th GOP starts. π_n is the probability of states in the hidden chain, conditioned on the observations available until time n .⁵

The next step in defining our problem is that of specifying criteria for the selection of a control sequence $(u_n)_{n \geq 0}$. Our goal is to maintain the receiver clock r_n slightly ahead of the transmitter clock t_n , by a fixed amount $d^* > 0$ (on average). For the system in state $[t, r, \pi]$, and for a control action u , we define the cost

$$c([t, r, \pi], u) = (r + T_d - t - \mathbb{E}_\pi\{f(u, \mathcal{H})\} - d^*)^2$$

$r - t$ is the difference between clocks before the application of control; after control, we expect that difference to become $r + T_d - (t + \mathbb{E}_\pi\{f(u, \mathcal{H})\})$. And we penalize deviations of this difference from a fixed value d^* , using a quadratic cost.

Finally, a Markov policy $\mathbf{g}_n : [t_n, r_n, \pi_n] \rightarrow u_n$ is obtained as a solution of

$$\min_{\mathbf{g}} \mathbb{E}^{\mathbf{g}} \left(\frac{1}{N} \sum_{n=0}^N c([t_n, r_n, \pi_n], u_n) \right)$$

Why is this a meaningful formulation of our “real-life” problem? To understand this, we study next properties of optimal policies.

4.1.2. Properties of an Optimal Controller

First of all, note that if r_n increases steadily faster than t_n , then unbounded large buffers are required at the receiver to hold data until its scheduled decoding time; and furthermore, one could attain better quality of the signal reconstructed at the receiver by increasing the number of packets allocated to each GOP. Conversely, if r_n increases slowly relative to t_n , eventually the receiver will starve. Hence, we see that the mean value $\mu^{\mathbf{g}} = \mathbb{E}^{\mathbf{g}}(r_n - t_n)$ of the (closed loop) clock differences should remain constant.

Now fix a constant value $d^* > 0$ by which r_n should remain ahead of t_n on average, and consider the “meaning” of the cost for N large. Since the HMM is stationary and \mathbf{g} is markovian, the differences $d_n = r_n - t_n$ are stationary too (for large n), and hence $\text{Var}(d_n)$ is independent of n (for large n). By the Laws of Large Numbers (LLN) we have that $\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^N c([t_n, r_n, \pi_n], u_n)$ can be expanded as

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^N [d_n - d^*]^2 + [T_d - \mathbb{E}_\pi\{f(u_n, \mathcal{H})\}]^2 \\ + 2[d_n - d^*][T_d - \mathbb{E}_\pi\{f(u_n, \mathcal{H})\}] \\ = \text{Var}(d_n) + (d^* - \mu^{\mathbf{g}})^2 + (T_d - \mathbb{E}_\pi\{f(u_n, \mathcal{H})\})[\dots] \text{ (a.c.)} \end{aligned}$$

There are two properties of an optimal \mathbf{g} that are of interest to us. First, it is clear that an optimal controller will make $\mu^{\mathbf{g}} = d^*$ –else the cost is increased by $(d^* - \mu^{\mathbf{g}})^2$ –, thus making the parameter d^* be what we intuitively want it to be: the average amount by which r_n stays ahead of t_n . Second, if in steady state we have $\mathbb{E}_\pi\{f(u_n, \mathcal{H})\} \neq T_d$, then clearly $\mu^{\mathbf{g}}$ cannot be a constant, and hence the first condition cannot be satisfied. So we see that for an optimal \mathbf{g} , the last two terms in the limit drop out. And furthermore, since for N large the cost becomes $\text{Var}(d_n)$, asymptotically the optimal policy yields a *minimum variance* controller. This is desirable because it allows us to pick small values of d^* : large values are undesirable because they result in large buffers at the receiver; however, if d^* is “too close” to zero (say, less than a standard deviation), then the probability of the receiver starving is high.

4.1.3. Design of a Controller

Our problem is one of optimal control of a Markov chain, and it is solvable using Dynamic Programming (DP) [10]: an optimal controller is constrained by the DP equations

$$V_n([t, r, \pi]) = \min_{u \in \{0, 1, \dots\}} c([t, r, \pi], u) + \mathbb{E}\{V_{n+1}(h([t, r, \pi], u))\}$$

(for $0 \leq n < N$), terminal cost $V_N([t, r, \pi]) = 0$, and where $h([t, r, \pi], u)$ denotes the state transition function defined by equations (1)-(3). The value of an optimal control u_n is then defined as the u attaining the minimum in V_n .

Whereas it is conceptually straightforward to implement an optimal controller (just write a computer program to solve the DP equations), in practice this task may prove to be rather involved. Instead, in this work we design a controller by *enforcing* the optimality condition that $\mathbb{E}_\pi\{f(u, \mathcal{H})\} = T_d$: we define a stationary controller by setting $\mathbf{g}[t, r, \pi] = u$, where u is chosen such that $\mathbb{E}_\pi\{f(u, \mathcal{H})\}$ is as close as possible to T_d . Note that if we are willing to consider approximations to the optimal controller, this is an intuitively natural choice: we want to inject a number of packets that we believe will take about T_d clock ticks to be transmitted. For details on its implementation, see [15].

4.2. Redundancy Allocation

Redundancy allocation refers to deciding how to set parameters of the coder which control the degradation to the reconstructed video signal in the presence of channel errors. Our solution to this problem proceeds along similar lines to that of the rate control problem, and hence due to space constraints we do not include it here. For details, see [15].

5. EXPERIMENTAL RESULTS

We present experimental results testing two different aspects of our proposed system: the ability of our controller to track changes in channel states, and the quality of the

⁵ π_n plays the role of an *information state* [10, Ch. 6.5]: it captures everything that is known about the unobserved state.

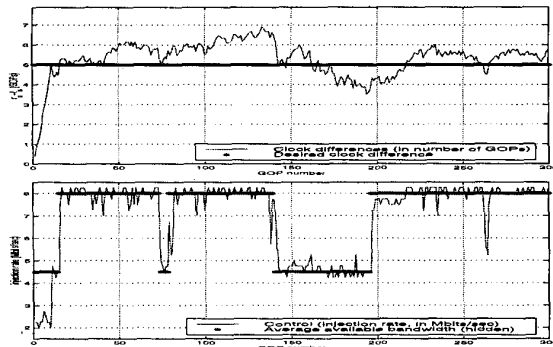


Figure 3: Performance of the proposed controller. Top: $r_n - t_n$, over 300 GOPs (5 minutes); bottom: sequence of control actions (i.e., injection rates) and (hidden) available bandwidth at the beginning of each transmission. Observe a few things: (a) setting $d^* = 5$ GOPs and a buffer of size 10 GOPs, it is more than enough to absorb all clock fluctuations; (b) the controller effectively tracks the available bandwidth even though it cannot be observed; (c) when transmission starts, there is a steady increase of $r_n - t_n$ up until it reaches the value of $d^* = 5$ GOPs: this is caused by a controller injecting less than the available bandwidth, to help build up a buffer at the receiver.

video signal delivered at the receiver. For this purpose, we define a simple channel: a hidden Markov chain with two states, “good” and “bad” [8]. Intuitively, in the good state many packets are sent with few losses; in the bad state, less packets are sent and with a higher loss rate. The average time it takes to write a packet of size 256 Kbits to a socket, on a fast Sparc20, is 0.0026675 seconds, resulting in 375 discrete-time ticks/sec. Observation probabilities are set to reflect loss rates of 5% and 10%, and injection rates of 8 and 4.5 Mbits/sec in the good and bad states respectively. The chain stays in any state with probability 0.9999, and the initial distribution is uniform.⁶ Selected results are shown in Figs. 3 and 4; many more can be found in [15].

6. CONCLUSIONS

The most important conclusion drawn from these results is that, as it is today, the public Internet provides bit pipes thick enough to support the transmission in real-time of video signals of significantly higher quality than is currently possible. We showed how minor modifications to existing transport protocols, a video coder resilient to packet losses, in conjunction with a carefully designed controller sitting at the interface between the network and the transmitter, result in a complete system capable of delivering video signals over the public network at TV-broadcast quality.

An important problem requiring further research is error-resilient video coding. The coder used in this work is a “toy”

⁶The parameters of the chain were chosen to match measurements made under high and low load conditions, between `lcavsun9.epfl.ch` (Switzerland) and `athena.cs.uiuc.edu` (US). For these preliminary experiments however, focused on the controller, this choice was done “by hand”: parameter estimation and measurement update issues are dealt with extensively in [14].

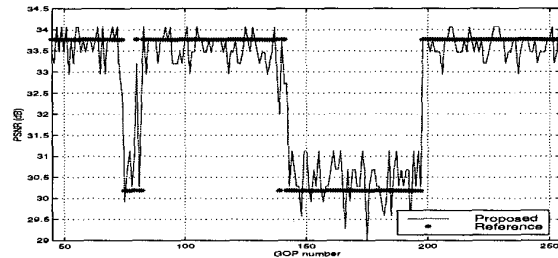


Figure 4: Average PSNR over a fixed GOP transmitted repeatedly over time, for the transmission in Fig. 3. Reference: MPEG-2, protected with a rate 0.65 code, and known bandwidth. The rate 0.65 is the highest that ensures recovery of *every* transmitted GOP in this example. Observe how our system can deliver video of quality comparable to that of current TV-broadcasts.

version, developed only for the purpose of having a complete testbed for our controller. Yet note how the power of our JSCC framework is illustrated by the fact that even this toy coder is able to attain performance comparable to that of a fully optimized standard plus good channel codes: this is among the best source/channel separation methods have to offer. We expect that a fully optimized error-resilient coder will significantly outperform the results reported here, and hence this is the main focus of our current research. We are also studying issues related to the smoothness of our control signals, to avoid large fluctuations of video quality.

Acknowledgements. To W. Turin, for educating us on the usefulness of HMMs as a tool to approximate complex stochastic phenomena. To M. Vetterli and D. Deatrich, for access to and thorough information on EPFL’s network.

7. REFERENCES

- [1] J. Boyce. Packet Loss Resilient Transmission of MPEG Video over the Internet. *Signal Processing: Image Communication*, 1999 (to appear).
- [2] A. R. Calderbank. The Art of Signaling: Fifty Years of Coding Theory. *IEEE Trans. Inform. Theory*, 44(6):2561–2595, 1998.
- [3] Z. Chen, S.-M. Tan, R. Campbell, and Y. Li. Real Time Video and Audio in the World Wide Web. In *WWW4*, 1995. Available from <http://www.vossic.com/>.
- [4] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., 1991.
- [5] G. Davis, J. Danskin, and X. Song. Joint Source and Channel Coding for Internet Image Transmission. In *Proc. Int. Conf. Image Proc. (ICIP)*, Lausanne, Switzerland, 1996.
- [6] D. Dwyer, S. Ha, J.-R. Li, and V. Bhargavan. An Adaptive Transport Protocol for Multimedia Communication. In *Proc. Int. Conf. Multimedia Computing and Systems*, Austin, TX, 1998.
- [7] M. W. Garrett. *Contributions Toward Real-Time Services on Packet Switched Networks*. PhD thesis, Columbia University, 1993.
- [8] E. N. Gilbert. Capacity of a Burst-Noise Channel. *Bell Syst. Tech. J.*, 39(5):1253–1265, 1960.
- [9] V. Jacobson. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM*, 1988.
- [10] P. R. Kumar and P. Varaiya. *Stochastic Systems: Estimation, Identification and Adaptive Control*. Prentice Hall, Inc., 1986.
- [11] S. R. McCanne. *Scalable Compression and Transmission of Internet Multicast Video*. PhD thesis, University of California, Berkeley, 1996.
- [12] Real Networks, Inc. *Real Video Player*. <http://www.real.com/>.
- [13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. *RTP: A Transport Protocol for Real-Time Applications*. RFC 1889, available from <http://www.ietf.org/>.
- [14] S. D. Servetto. Hidden Markov Modeling of TCP Channels. Private manuscript, June 1999. In preparation for publication.
- [15] S. D. Servetto and K. Nahrstedt. Broadcast-Quality Video over IP. Submitted for publication. Available from <http://www.ifp.uiuc.edu/~servetto/publications/>.
- [16] S. D. Servetto, K. Ramchandran, V. A. Vaishampayan, and K. Nahrstedt. Multiple Description Wavelet Based Image Coding. Submitted for publication. Available from <http://www.ifp.uiuc.edu/~servetto/publications/>.
- [17] W. Stevens. *TCP/IP Illustrated, Vol. 1: The Protocols*. Addison Wesley, 1994.
- [18] W. Turin and R. van Nobelen. Hidden Markov Modeling of Flat Fading Channels. *IEEE J. Select. Areas Commun.*, 16(9):1809–1817, 1998.